

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer readable media having instructions stored thereon that, when executed by a processor, causes the processor computer-enabled system to provide a network-accessible service, the instructions comprising:

an annotated source code, which is a programming language augmented with declarative meta-data that capable of exposing program logic as a network-accessible service;

at least one deployed service component that capable of providing the network-accessible service to a client; and

an enhanced compiler that capable of analyzing the annotated source code, recognizing-recognizes numerous types of meta-data annotations, and generating-generates a mechanism, which can include one or more of: object files, software components and deployment descriptors, to facilitate the deployment of the at least one service component, wherein the enhanced compiler operates to automatically create, deploy and manage at least one security type using a specification provided by the annotated source code, wherein the at least one security type is can be applied to at least one of: request messages, callbacks, response messages, and requests message and/or callbacks originating with an intermediate service in a chain.

2. (Currently Amended) The computer readable media system according to claim 1, wherein:

the network-accessible service is a Web service.

3. (Currently Amended) The computer readable media system according to claim 1, wherein:

the network-accessible service system simultaneously manages multiple transactions, wherein each transaction can be a conversation of a request and/or a response from the client for the network-accessible service.

4. (Currently Amended) The computer readable media system according to claim 3, wherein:

the network-accessible service system manages multiple asynchronous transactions, wherein within each asynchronous transaction, the response may be temporally separated from the initiating request for the network-accessible service from the client.

5. (Currently Amended) The computer readable media system according to claim 1, the instructions further comprising:

an integrated development environment (IDE) facilitating a graphical interface-based design and deployment of the network-accessible service.

6. (Currently Amended) The computer readable media system according to claim 1, wherein:

the annotated source code is Java-based object oriented programming language based.

7. (Currently Amended) The computer readable media system according to claim 1, wherein:

the meta-data can be either in-file with the annotated source code, or in a separate file, which can be a specially formatted XML file.

8. (Currently Amended) The computer readable media system according to claim 1, wherein:

the annotated source code facilitates access to an external service, which can be one of stateful, stateless, synchronous, and asynchronous.

9. (Currently Amended) The computer readable media system according to claim 1, wherein:

the annotated source code defines a wire binding between the network-accessible service and a physical wire format and/or protocol.

10. (Currently Amended) The computer readable media system according to claim 9, wherein:

the wire binding can be at least one of:

SOAP over HTTP or SMTP;

transport of XML via generic HTTP Post;

transport of XML over other protocols such as FTP and mail;

and transport of XML over messaging services such as JMS or MSMQ.

11. (Currently Amended) The computer readable media system according to claim 2, wherein:

the at least one service component comprises a servlet container and an ~~Enterprise~~enterprise server-side managed component container, which are coupled together to deploy a Web service.

12. (Currently Amended) The computer readable media system according to claim 11, wherein:

the servlet container is capable of ~~at least one of~~comprises:

listening and responding to a service request from the client; and

identifying and queuing the service request to be buffered.

13. (Currently Amended) The computer readable media system according to claim 11, wherein:

the ~~Enterprise~~enterprise server-side managed component container dispatches a service request based on meta-data to a stateless or stateful component.

14. (Canceled).

15. (Currently Amended) The computer readable media system according to claim 1, wherein:

the at least one security type is comprised of at least can be one of:
transport security for a data communications channel;
message based encryption security to protect headers and payloads in messages;
authentication of users or computing devices for channel or message based security;
digital signatures for verifying the originator of a message; and
user identity and declarative role-based security for granting or deny a client access to a particular service.

16. (Currently Amended) The computer readable media system according to claim 1, wherein:

the enhanced compiler creates reliable messaging software for the network-accessible service using a specification provided by the annotated source code, wherein the reliable message software ~~is capable of guaranteeing~~ message delivery for communication between the service and the client.

17. (Currently Amended) The computer readable media system according to claim 1, wherein:

the enhanced compiler creates an interceptor using a specification provided by the annotated source code, wherein the interceptor ~~is capable of~~ allowing at least one user-defined pre or post processing method to be applied to a message from the client.

18. (Currently Amended) The computer readable media system according to claim 17, wherein:

the at least one processing method ~~can be~~ comprises at least -one of:
interception and transformation of message headers and contents;
message logging and auditing;
message redirection; and
protocol conversion.

19. (Currently Amended) A computer-enabled method to provide a network-accessible service, comprising:

exposing program logic as a network-accessible service using an annotated source code, which is a programming language augmented with declarative meta-data;

providing the network-accessible service to a client via at least one deployed service component; and

analyzing the annotated source code, recognizing numerous types of meta-data annotations, and generating a mechanism that comprises, ~~which can~~

include one or more of: object files, software components and deployment descriptors, to facilitate the deployment of the at least one service component; automatically creating, deploying and/or managing at least one security type using a specification provided by the annotated source code, wherein the at least one security type is applied to at least one of ~~can be applied to one of~~: request messages, callbacks, response messages, and requests message and/or callbacks originating with an intermediate services in a chain.

20. (Original) The method according to claim 19, wherein:
the network-accessible service is a Web service.

21. (Original) The method according to claim 19, further comprising:
simultaneously managing multiple transactions, wherein each transaction is a conversation of a request and/or a response from the client for the network-accessible service.

22. (Original) The method according to claim 21, further comprising:
managing multiple asynchronous transactions, wherein within each asynchronous transaction, the response may be temporally separated from the initiating request for the network-accessible service from the client.

23. (Original) The method according to claim 19, further comprising:

facilitating a graphical interface-based design and deployment of the network-accessible service.

24. (Currently Amended) The method according to claim 19, further comprising:

facilitating access to an external service that comprises at least one of,
which can be one of stateful, stateless, synchronous, and asynchronous.

25. (Original) The method according to claim 19, further comprising:
defining a wire binding between the network-accessible service and a
physical wire format and/or protocol.

26. (Original) The method according to claim 20, further comprising:
listening and responding to a Web service request from the client; and
identifying and queuing the Web service request to be buffered.

27. (Original) The method according to claim 20, further comprising:
dispatching a Web service request based on meta-data to a stateless or
stateful component.

28. (Canceled).

29. (Currently Amended) The method according to claim 19, further comprising:

creating reliable messaging software for the network-accessible service using a specification provided by the annotated source code, wherein the reliable message software is capable of guaranteeing message delivery for communication between the service and the client.

30. (Currently Amended) The method according to claim 19, further comprising:

creating an interceptor using a specification provided by the annotated source code, wherein the interceptor is capable of allowing at least one user-defined pre and post processing method to be applied to a message from the client.

31 – 46 (Cancelled).

47. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a computer-enabled system to:

~~exposing~~ expose program logic as a network-accessible service using an annotated source code, which is a programming language augmented with declarative meta-data;

~~providing~~ provide the network-accessible service to a client via at least one deployed service component; and

~~analyzing~~ analyze the annotated source code, recognizing numerous types of meta-data annotations, and generating a mechanism, which can includes at least one or more of: object files, software components and deployment descriptors, to facilitate the deployment of the at least one service component;

~~automatically creating~~ create an interceptor using a specification provided by the annotated source code, wherein the interceptor is ~~capable of allowing~~ at least one user-defined pre or post processing method to be applied to a message from the client.